# Learned Traversability Priors for Visual Navigation

Will Huey
*Cornell University*
Ithaca, NY
wph52@cornell.edu

Sean Brynjólfsson
*Cornell University*
Ithaca, NY
smb459@cornell.edu

## 1. Introduction

A key problem in mobile robotics is navigation in new or dynamic environments. Recently, there has been extra focus on visual navigation, where the robot must navigate using only image observations. Images are rich enough to characterize a large number of potential traversals, and there are some visual semantic cues about traversability that geometric information does not provide. Tall grass may be traversable, but would be captured as an obstacle by point cloud based methods. Wet concrete may only be distinguishable from regular concrete by construction tape or cones in the vicinity, which is a semantic clue that requires vision to understand.

A navigation model must have a high level understanding of the environment to perform search, as well as the capability to avoid collisions. These are two fundamentally different problems, but they are rarely treated as such in visual navigation literature. We hypothesize that providing a traversability prior to visual navigation models can improve performance on out of distribution scenarios, especially when there is limited training data. In this paper, we demonstrate that weak traversability priors can be obtained from large open vocabulary image segmentation models. We then apply model distillation techniques to train a smaller traversability prediction network capable of real time inference, and demonstrate a heuristic that uses this network to perform obstacle avoidance. Finally, we investigate the use of these models for navigation in a behavioral cloning setting.

## 2. Related Work

Navigation methods are broadly centered around two categories: geometric and visual. Geometric approaches construct a geometric representation of the world from depth cameras, lidar, and robot odometry. For ground robots, this is often an elevation map [13], both for ease of implementation and in order to reduce roll-out computation costs. Once a geometric representation of the world has been produced, a heuristic [4] or learned-function [22] is then used to evaluate the traversability at each location on the map. Then, when provided a goal location, a path-finding algorithm like A* is used to find the path that minimizes the traversability cost. Some recent approaches also target a more end-to-end pipeline in which the planning itself is learned and can function with as little as one depth measurement [24].

Recently, visual navigation has become more popular. Lidar units are relatively large, expensive, and power hungry compared to cameras; different lidar scanners produce scans which have significantly distinct statistics which make methods hard to generalize; and real point cloud data is much more sparse and inconsistent in quality and size. Additionally, the advances in image semantic analysis can be incorporated into navigation pipelines. Images can capture semantic information about an environment that is not captured by depth.

The visual navigation task can be formulated as the following Markov Decision Process (MDP): given a set of recent images and robot state information, predict the next action for the robot to perform. The reward is task dependent, but it is a general measure of navigational performance. It may be defined by reaching some goal location, learning more information about the environment, or following a command. Many recent visual navigation methods have used elements of imitation learning to solve this MDP, taking advantage of the growing amount of publicly available robot rollout data.

Schmid, et al. and Frey, et al. used self-supervised learning to predict a mask of traversable areas on an image given a small number of initial expert demonstrations in a new environment [5] [17]. However, this method still requires projection of the traversability masks onto an elevation map. Dhruv, et al. applied offline imitation learning techniques on existing datasets of rollouts on varied robots and environments to create a generalized, end-to-end visual navigation model [19]. Within the aforementioned framework, ViNT [20] integrated a topological-graph based planner to enable global planning and exploration.

These methods show promise as robot-agnostic founda-

tion models for navigation. However, they are trained in a behavioral cloning-like setting, making them susceptible to covariate shift.

Another approach to learn traversability involves Vision-Language Models (VLMs). Vision-Language Models are an emerging class of multimodal architectures designed to operate over images and text. Since some abstract concepts are easier to verbalize than picture, VLMs provide an inherent benefit on image semantic analysis. Many recent models have been utilizing Contrastive Label-Image Pairs (CLIP) embeddings in order to make the bridge between text and images; CLIP is a shared latent space between image and text [15]. OVSeg (Open-Vocabulary Segmentation) is an segmentation model fine-tuned on the CLIP embedding space [12]. OVSeg provides for segmentation of arbitrary text input in a zero-shot setting.

In previous work, we used OVSeg as alternative method of generating a traversability signal, finding that it is possible to not only detect the abstract concept of 'traversability' in images, but also isolate which pixels in the image were labeled traversable.

## 3. Traversability Distillation

Traversability depends on a robot's capabilities, so any traversability prior will necessarily lack some knowledge that would be required for a general navigation model. Nevertheless, there are commonly traversable and untraversable terrains. For the vast majority of mobile robots, walls and obstacles cannot be traversed, but smooth ground is generally traversable. During previous experimentation, we found that OVSeg, an off-the-shelf open vocabulary image segmentation model [12] is able to effectively label regions in an image according to abstract descriptions like "a traversable region" or "something a robot could walk on." This provides a zero-shot estimation for traversability in any environment. We propose using these predictions as a traversability prior. This avoids the problem of hand labeling many images, instead leveraging the knowledge and reasoning capabilities of a large transformer-based model.

Our approach is a simple and practical method targeted towards existing visual navigation architectures. Thus, it needs to be both light and fast enough to run in real time on limited compute. Using OVSeg for a constant set of labels leaves much to be desired because all of the overhead for open vocabulary is wasted on constant labels. Upon initial tests, OVSeg and other existing open vocabulary image segmentation methods are not fast enough; even OVSeg's smallest model has an inference time exceeding 5 seconds when running on a Jetson processor. This is unacceptable for real-time edge applications. Thus, we first aim to boil off all the unnecessary fat by distilling OVSeg over a constant prompt. To achieve better performance, we adopt a teacher-student model for distillation on traversable and untraversable predictions.

Hinton, et al. proposed Knowledge Distillation, in which the temperature adjusted KL divergence between the softmax probabilities of a teacher and a student model is minimized [8]. It was shown that, for student models that are not large enough to capture all the knowledge in the teacher model, it is best to use an intermediate temperature value (a temperature of between 2.5 and 20 worked best for networks of varying sizes on MNIST).

In practice, we choose to use the binary cross entropy loss instead of the KL divergence for knowledge distillation. Assuming that the entropy of the training set is constant, the minimizations of these expressions converge. The reasoning for using the cross entropy loss is as follows: let $D_h$ be the cross entropy loss, $D_{KL}$ be the KL (Kullback–Leibler) divergence, p(x) be the target distribution, and q(x) be the model distribution.

$$D_h = -\sum_{x \in X} p(x) \log q(x)$$

$$D_{KL} = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = D_h + \sum_{x} p(x) \log p(x)$$

$$D_h = D_{KL} + S_p$$

$S_p$ is the entropy of the target distribution X. If this is constant, then there is no difference in the minimization of these expressions. However, we train with minibatches of size of 8, so $S_p$ is almost certainly not constant across batches. Thus, the cross entropy loss is more robust in this situation.

Here, OVSeg is used as the teacher model and distilled to two different types of student models using a temperature of 4. Even though OVSeg is a generalist model, it has been evaluated to perform on-par with task-specific models like FCN, Deeplab, and SelfTrain [12]. This suggests that OVSeg is among the best source models available for distillation because it may match the performance of a dedicated model specifically for our target, traversability.

Two student models were investigated. The first is a pretrained FasterViT backbone [7] with a feature pyramid network as described in He, et al. [11] to obtain image segmentations. We use the smallest pretrained FasterViT model (31.4M params, pretrained on Imagenet 1k). FasterViT was chosen because it has achieved the current Pareto-front with respect to throughput vs. accuracy on a wide range of benchmarks. Although not explicitly designed for semantic segmentation tasks, it beats similar models on ADE20K. Additionally, because it consists of two convolutional layers followed by two transformer layers, visual prompt tuning can be used to accelerate training speeds [10]. The only learnable parameters are visual prompts inserted between each transformer layer (accounting for less than 1% of total model parameters), the FPN transformations, and a final

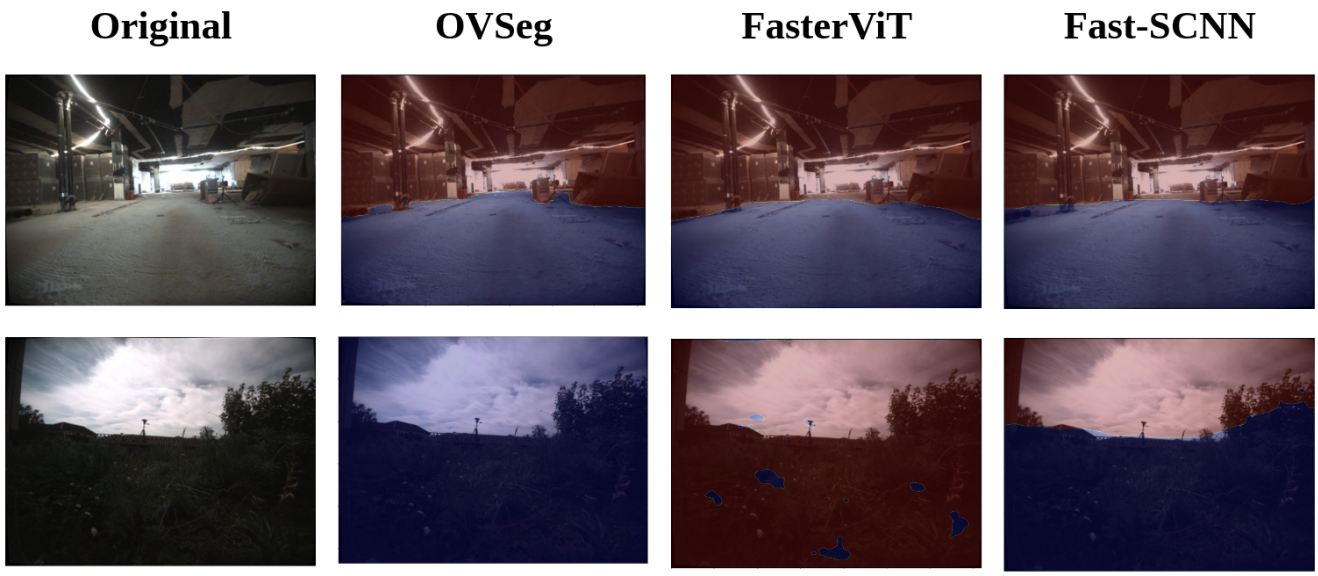


Figure 1. Examples of two images in the test set and their traversability segmentations by OVSeg and the two student models trained on the RECON, KITTI, and UTIAS datasets. The top row shows an image on which OVSeg performs very well, and the other models are able to reasonably replicate its predictions. The bottom row shows an image that OVSeg fails on. Interestingly, the FasterViT student model fails in the manner, but almost predicts the exact opposite signal. Meanwhile, Fast-SCNN is able to create a reasonable prediction, assuming the robot can walk through tall grass.

fully connected layer. This allowed us to quickly fine tune a relatively large student model, training only 867K parameters. The second student model is a fully fine tuned Fast-SCNN model pretrained on the Cityscapes dataset. [14]. Since the Cityscapes dataset contains many of the features that correspond to traversability (having separate segments for roads, sidewalks, obstacles, etc.), we hypothesized that this model's parameters would already encode some representation of traversability, allowing it to converge faster and generalize better. FastSCNN only has 1.5M parameters, but all of these needed to be tuned.

Following [19], we combine a variety published datasets containing egocentric images taken by different robots in varying terrain. Samples were selected from these datasets to increase the diversity of the images in the training set. For example, images in KITTI are captured at 10hz, which results in a large number of extremely similar images. Since the traversability model does not depend on sequences of images or commands, many of these examples could be removed. To test the models, we reserved a test set of 436 images taken by an Anymal D robot navigating a construction site and an office space. The camera intrinsics, robot size, orientation of the camera with respect to the robot, image resolution, and environment of the test set is significantly different from the training set. This is important because the goal of the traversability prediction network is to generalize to different sensors and robots. Further information about the specific datasets and the number of images selected from them is given in Table 1.

The teacher is given the prompts "something a robot could walk on" and "other," and the student is trained on the resulting segmentation masks. We apply a set of templates to these prompts (such as: "an image of" or "a dark photo of") and average the results, as performed by the authors of OVSeg. Initially, other similar prompts were also evaluated, such as "traversable terrain" and "obstacles." Since there is no existing dataset of images with pixel-level traversability labels, the prompts cannot be empirically evaluated or numerically optimized. In fact, creating such a dataset would be impossible, because traversability is dependent on the robot and somewhat subjective. Thus, the prompts "something a robot could walk on" and "other" were based on qualitative observations of the outputs of OVSeg. Specifically, we looked for performance on challenging features (such as stairs or ramps), frequency of catastrophic failures (where the entire mask would be traversable or untraversable), and the granularity of the masks.

The models were trained for 25 epochs with a learning rate of 6e-4, which took between 1-4 hours depending on the model. These parameters were found using grid search. In all cases, images were interpolated to a standard size. The SaCSON dataset contains 120x160 images, which needed to be upsampled for use with the pretrained FasterViT model. For each model type, we tested training with all of the training datasets, and with all except for SaCSON. The performance of the four different architecture and train-

| Dataset | Platform | Samples | Env. |
|---|---|---|---|
| SACSoN [9] | TurtleBot2 | 12512 | office |
| RECON [18] | Jackal | 3912 | off-road |
| KITTI [6] | Car | 3301 | self-driving |
| UTIAS [3] | Grizzly | 22 | off-road |
| RSL Lab | Anymal D | 275 | busy office |
| Schulstrasse 44 | Anymal D | 161 | construction |
| Total | | 20183 | |

Table 1. The datasets used for traversability knowledge distillation from OVSeg. These cover a wide range of environments that may be encountered by mobile robots, including offices, trails, sidewalks, construction sites, and roads.

ing set combinations is shown in Table 2. We report the mean intersection-over-union (mIOU) of the distilled network with the teacher network for our robot-acquired images. Although SaCSON contributed many additional training points (including points within an office space, which was similar to one of the test sets), performance was better for both model architectures without it. Our experiments were on photos of size 224x224, so we did not need to worry about extremely low resolution images. For robots with low resolution cameras, it is possible that the models trained on the full data would be better.

Figure 1 shows examples of two different images in the test set. Despite the high resolution training sets containing no indoor images, these models performed remarkably well on the test set. The student models were able to predict that walls are untraversable, having only seen images taken by vehicles in various outdoor terrain. However, their segmentations are missing many of the fine grained features of OVSeg (such as the column on the left). This is most likely due to issues with the model resolution. Increasing model resolution without a significant penalty to runtime would help boost performance.

The bottom row shows an image that OVSeg fails on. There are shadows, the scene is heavily occluded by grass, and it suffers from glare. It predicts that the entire image is untraversable, which is a behavior that occasionally occurs for difficult or ambiguous images. This demonstrates the importance of the open vocabulary method that is chosen as the teacher method. For our purposes, failures like this can happen occasionally, as long as the model is correct on average.

We also report of the latency of the models running on an Nvidia A6000 GPU with a batch size of 1. For robotics applications, images often need to be processed in sequence, so the single image latency is more important than model throughput with large batches. Before distillation, the model could not be run in real time on a workstation, let alone on a robot with limited compute. Using Fast-SCNN, we achieved a speedup of over 400x.

| Model | Dataset | mIoU | Latency (ms) | Speedup |
|---|---|---|---|---|
| OVSeg | - | 0.99 | 1651.9 | 1x |
| FasterViT | full | 0.75 | 18.21 | 91x |
| FasterViT | partial | 0.80 | 17.68 | 93x |
| Fast-SCNN | full | 0.79 | 4.11 | 402x |
| Fast-SCNN | partial | 0.72 | 4.03 | 409x |

Table 2. Model performance metrics. Models with the full label were trained on SaCSON, RECON, KITTI, and UTIAS images. Models with the partial label were trained only on RECON, KITTI, and UTIAS.

## 4. Heuristic Guided Navigation

We evaluated our models in Nvidia Isaac Sim, a robotics simulator built on top of Nvidia's Omniverse. Isaac Sim supports photorealistic rendering and provides an API to simulate a variety of mobile robots. For these experiments, we use an ANYbotics ANYmal C. We obtained high-quality photogrammetry meshes from SketchFab. Our simulated experiments make heavy use of the Lincolns Inn Chapel Undercroft and Lake Shore Drone Scan [1, 2]. No modifications other than converting the scan into the .usd file format and applying a physics material took place. However, it must be noted that the conversion process did degrade the mesh and textures. This was acceptable to us because the input to our model is an image at a resolution of 224x224, so losing out on some of the high frequency details should be relatively insignificant.

The distilled traversability network is fast enough to run in real time, and its predictions match the teacher with good accuracy. However, since the teacher model's outputs are based on natural language prompts, and there are no ground truth traversability masks, it is impossible to tell whether these predictions are good enough to provide a meaningful prior to a navigation model. As a proof of concept, we implemented four simple heuristics that allow the robot to explore and avoid obstacles autonomously using only the current traversability prediction as input.

1. **Eight-Column**. The eight column approach breaks the image into eight 224x28 pieces. Let $P_0 ... P_7$ be the individual pieces and let $P_{\max}$ be the most traversable piece. In the column approach, the robot tended to get too close to the columns because it would continue straight forward even if a large portion of one of its sides was untraversable.

$$\text{action} = \begin{cases} \text{left}, & P_{\max} \in P_0, P_1, P_2 \\ \text{straight}, & P_{\max} \in P_3, P_4 \\ \text{right}, & P_{\max} \in P_5, P_6, P_7 \end{cases}$$

Figure 2. The four primary heuristics we tested out on our traversability signals rolled out ten times each on the Lincoln Inn Chapel Undercroft environment. **Leftmost:** Columns. **Center left:** Octants. **Center right:** Bottom-Heavy Octant. **Rightmost:** Bottom-Heavy Pairwise Octant. See Section 6 for more information on the right turning bias present in these rollouts.

2. **Octant**. The octant approach breaks the image into eight 112x56 pieces, so two rows and four columns. $P_0, P_1, P_2, P_3$ are in the top row from left to right, $P_4, P_5, P_6, P_7$ are in the bottom row also from left to right. With the octant approach we intended to force the robot to pick a direction instead of being able to continue straight-ahead. Column avoidance did improve, but since the bottom half of the camera image represents points quite close to the robot, the robot would not preemptively try to navigate around columns.

$$\text{action} = \begin{cases} \text{left}, & P_{\max} \in P_0, P_4, P_5 \\ \text{straight}, & P_{\max} \in P_1, P_2 \\ \text{right}, & P_{\max} \in P_3, P_6, P_7 \end{cases}$$

3. **Bottom-Heavy Octant**. The bottom heavy octant approach breaks the image into four 56x56 pieces along the top row and four 186x56 pieces along the bottom row; still two rows and four columns like in the Octant case with pieces labeled the same. See Figure 3. By including the bottom three-quarters of the image, we expected the robot to display more avoidant responses to columns in the distance. We observed this behavior but along with it came "farsightedness", where the robot seemed to disregard its local traversability to pursue a high traversability signal far away, often taking sharp turns and narrowly passing columns to do so.

$$\text{action} = \begin{cases} \text{left}, & P_{\max} \in P_0, P_4, P_5 \\ \text{straight}, & P_{\max} \in P_1, P_2 \\ \text{right}, & P_{\max} \in P_3, P_6, P_7 \end{cases}$$

4. **Bottom-Heavy Pairwise Octant**. This begins by breaking apart the image in the same way as the Bottom-Heavy Octant, but now we group the pieces into pairs of neighbors along the columns. By introducing pairs, we reintroduced a straight command to the bottom row of pixels, hoping that it might allow the robot to take a more consistent trajectory. This once again seemed to suffer the same apathy towards obstacles in the periphery as the Eight-Column approach, however.

$$\text{action} = \begin{cases} \text{left}, & (P_i, P_j)_{\max}, \in (P_0, P_1), (P_4, P_5) \\ \text{straight}, & (P_i, P_j)_{\max} \in (P_1, P_2), (P_5, P_6) \\ \text{right}, & (P_i, P_j)_{\max} \in (P_2, P_3), (P_6, P_7) \end{cases}$$


Figure 3. A demonstration of the robot's perception overlaid with both the traversability mask (in red) and the most-traversable octant (in green). In this case, the robot predicts to turn left using the Bottom-Heavy Octant heuristic.

With these simple heuristics, we demonstrate that our traversability signal provides meaningful and actionable information for autonomous navigation of environments which the robot has *never* seen before, see Fig. 2. Even

further, our method does not need to construct a 3D representation of the world with which to perform point queries or other such algorithms to determine where it can go, keeping it both light-weight and decoupled from any particular representation of the world.

The primary downside to our approach is that we do not implement a backtracking behavior (in fact, there is no input to any of the heuristics which tells the robot to back up). This presents challenges in narrow passageways since the heuristic will gladly guide the robot into dead ends. This is intentional: the robot should not be expected to do any better without a more comprehensive approach. Backtracking and avoiding dead-ends requires a stateful working knowledge of the environment which cannot be deduced from a single image alone. We regard this problem as distinct from traversability estimation because it involves context beyond the image.

Another downside to this heuristic out-of-box is that there is no clear way to decide between two good avenues—sometimes when faced with a fork in the road, the robot vacillates between traversable options and is unable to make progress in any direction. Failing in this manner though is itself promising because it warrants consideration of ways to fuse this approach with another that can instill a global objective. If there are two or more options detected as traversable, then the traversability estimator has done its job.

Since a major advantage to vision-based traversability is its ability to respond to semantic information that may confound geometric models, we conducted some qualitative assessments of our traversability estimator on features on water. Specifically, we chose a calm lake which cannot be distinguished from a smooth surface by geometry alone (see Fig. 4. In short, our traversability estimator is weary of water but will suggest it over a blatant obstacle. We tested the Bottom-Heavy Octants heuristic out on various locations of the Lake Shore Environment. We note that there seems to be a peninsula effect whereby traversability is higher for an outcrop of land surrounded by water or land with a physical barrier on one side and water on the other. This causes the robot under our heuristic to enter a dead-end scenario. However, once the robot gets sufficiently close to the water, the traversability score of the water tends to increase, suggesting that our traversability estimator may not unequivocally treat water as untraversable.

## 5. Traversability Priors for Existing Models

The heuristic approach is a useful proof of concept, but it does not help boost the capabilities of existing learning-based techniques for navigation. Since many visual navigation models currently use training in the style of imitation learning, we specifically chose to investigate the effect of adding a traversability prior on the performance of behavioral cloning models (a specific type of imitation learning).

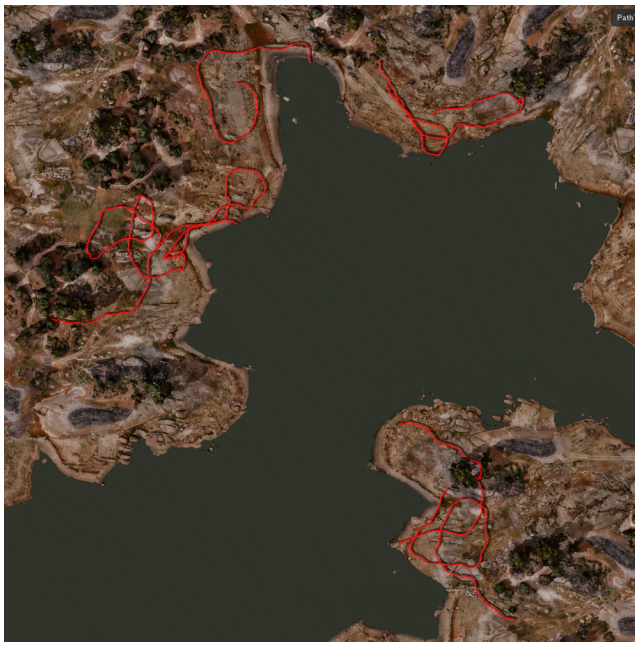


Figure 4. An aerial view of the Lake Shore Drone Scan [2] showing several autonomous roams along the coastline.

Figure 5. Three traversability segmentations and heuristic selections with water in the vicinity. The two on the right exhibit the peninsula effect.

First, we would like to provide a mathematical intuition for why a traversability prior may help improve performance. Let $J_\pi$ be the cost-to-go of the current policy, $V_{\pi_E}$ be the cost-to-go of the optimal policy (expert), T be the task horizon, and $l(s, \pi)$ be the surrogate loss function that is optimized in place of the value function (since the value function is not fully known). Ross & Bagnell showed that behavioral cloning results in quadratic compounding of error with the task horizon [16]:

$$J_\pi \leq J_{\pi_E} + T^2 \mathbb{E}_{s\ d_{\pi_E}}[l(s, \pi)]$$

Substituting the value function V for the negative cost-to-go, it follows that:

$$|V_\pi - V_{\pi_E}| \leq T^2 \mathbb{E}_{s\ d_{\pi_E}}[l(s, \pi)]$$

To mitigate this problem, Ross & Bagnell proposed DAgger, a no-regret online learning algorithm that results

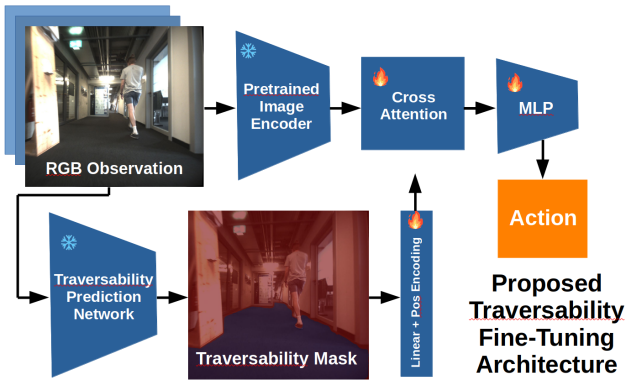


Figure 6. The proposed method for fine tuning an existing image navigation model on traversability data. Traversability masks are produced by the distilled open vocabulary segmentation network. The MLP head will depend on the downstream task (for example, it may predict explicit actions or normalized waypoints).

in only linear error compounding of error. DAgger requires expert intervention during training rollouts, which is not feasible for navigation models that aim to generalize to any robot and environment.

The other way to avoid the effects of covariate shift is to gather training data that covers a larger subset of the possible states. Let $R_{\max}$ be the maximum possible reward, $\gamma$ be a discount factor, and $D_{TV}(\rho_\pi, \rho_{\pi_E})$ be the state-action distribution discrepancy between the learner and the expert. Xu, et al. derived the following error bound for the value gap in behavioral cloning [23]:

$$|V_\pi - V_{\pi_E}| \leq \frac{2R_{\max}}{1-\gamma} D_{TV}(\rho_\pi, \rho_{\pi_E}).$$

Thus, if the state-action distribution discrepancy is minimized, this can result in a tighter bound on the value gap and a better model. One way to decrease this discrepancy is to provide the learner with actions in a wider subset of the possible states. Our key assumption is that, by adding a traversability prior, the space of possible states is constrained, thus decreasing the state-action discrepancy. For example, obstacles that would previously correspond to entirely different states (such as an image of a large column vs. an image of a human) simply become untraversable terrain given the prior.

To explicitly provide traversability information to these models, we apply a cross attention mechanism [21] on the feature maps of the RGB image and the tokenized traversability mask as depicted in Figure 6. The RGB feature maps are represented as the decoder signal to the attention mask (providing queries), and the traversability masks as the encoder signal (providing keys and values). This allows the feature map to attend to each patch in the traversability mask, so it can learn which parts of the mask

Figure 7. An example of an expert run in the environment of the Lincoln Inn Chapel Undercroft (simulated in Omniverse's Isaac Sim); six poses equally spaced were selected from one of the expert runs.

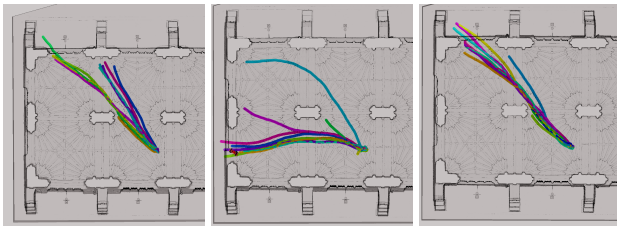

Figure 8. Comparison of the behavior cloning ablations. **Left:** Image only. **Center:** Traversability only. **Right:** Fusion.

are relevant. Additionally, a residual connection is added from the RGB feature maps to the cross attention output.

To test out behavior cloning with the traversability mask, we picked the Lincoln Inn Chapel Undercroft [1] since its rows of columns were consistent and there were otherwise few sporadic features to interfere with the robot. We designated a spawn location and then rolled out 5 different expert-guided rollouts of 150 image-action pairs. The action space consisted of a rotation command, and it was assumed that the robot would always walk forward with constant velocity. In the expert runs, the robot began facing NW at the central pillar, turned right to avoid it, then left until it was walking towards the central pillar on the western wall. In test runs, the robot was placed in a different location and faced a different pillar, in order to test generalization ability. We attempted two ablations: one only using images, and one only using the traversability signal. The results after training are shown in Fig. 8. The image only model and fusion model produced similar results. They were able

to copy the expert in veering right around the column, but failed to subsequently turn left. In general, these models do not appear to replicate the expert actions in any meaningful way. This is probably due to the severely limited size of the behavioral cloning training set, as compared with the size of the model.

## 6. Future Work

One limitation of the proposed cross attention mechanism is its quadratic runtime complexity. In these experiments, images were represented as sequences of at most 49 patches, which was small enough to avoid a significant effect on the runtime of the model. However, for higher resolution images and priors, it may become important to use an approximation of the full attention matrix.

We originally intended to use a real wheeled robot for our experiments, but we ran into various hardware dependencies and software version issues that made it impossible to run our model on the robot. The Isaac Sim environments we used are based on real point clouds, and the 224x224 images are highly photorealistic. Since the traversability prediction networks were trained using real images, sim-to-real transfer for the model will not be a problem. However, in order to more rigorously investigate the effect of traversability predictions on network performance, it is important to obtain real life results. In fact, due to limitations of the Anymal locomotion policy in simulation, we experienced a right-turn bias across all of our rollouts—even in symmetrical environments. This is not a result of a training bias because we flipped all inputs randomly along the vertical axis after segmentation was performed by the teacher model. This rightwards skew actually results from a bias in the simulated locomotion policy. In simulation, our robot has a leftwards tilt, meaning that the ground on the right side of the image is usually higher than on the left, this causes the right-half of the image to appear more traversable on average. In real life, with a more robust locomotion controller and data that is more similar to the training distribution, the results may actually be better.

Finally, it would be interesting to see how existing visual navigation models (such as GNM and Wild Visual Navigation) perform when provided with traversability signals. The behavioral cloning presented here did not demonstrate any compelling results, but it used extremely limited data on a small and unrealistic task. Given that a simple heuristic on top of the traversability prior is able to achieve consistent obstacle avoidance, coupling it with a model that can also take into account global information should result in good navigation performance.

## References


[1] artfletch. Lincoln's inn chapel undercroft. https://skfb.ly/oItqQ. Licensed under Creative Commons Attribution 4.0 International. 4, 7

[2] Per Bellersen. Lake Shore Drone Scan. https://skfb.ly/oDrNw, 2022. Licensed under Creative Commons Attribution 4.0 International. 4, 6

[3] Lee Clement, Mona Gridseth, Justin Tomasi, and Jonathan Kelly. Learning matchable image transformations for long-term metric visual localization. *IEEE Robotics and Automation Letters*, 5(2):1492–1499, 2020. 4

[4] Tung Dang, Marco Tranzatto, Shehryar Khattak, Frank Mascarich, Kostas Alexis, and Marco Hutter. Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics*, 37(8):1363–1388, 2020. Wiley Online Library. 1

[5] Jonas Frey, Matías Mattamala, Nived Chebrolu, Cesar Cadena 0001, Maurice F. Fallon, and Marco Hutter 0001. Fast traversability estimation for wild visual navigation. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *Robotics: Science and Systems XIX*, 2023. 1

[6] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 4

[7] Ali Hatamizadeh, Greg Heinrich, Hongxu Yin, Andrew Tao, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. Fastervit: Fast vision transformers with hierarchical attention. In *Submitted to The Twelfth International Conference on Learning Representations*, 2023. Under review. 2

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 2

[9] Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Sacson: Scalable autonomous control for social navigation. *Robotics and Automation Letters (RA-L)*, 2023. 4

[10] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[11] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[12] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[13] Takahiro Miki, Lorenz Wellhausen, Ruben Grandia, Fabian Jenelten, Timon Homberger, and Marco Hutter. Elevation mapping for locomotion and navigation using gpu. In *International Conference on Intelligent Robots and Systems (IROS)*, 2022. 1

[14] Rudra P K Poudel, Stephan Liwicki, and Roberto Cipolla. Fast-scnn: Fast semantic segmentation network, 2019. 3

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. 2

[16] Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. 6

[17] Robin Schmid, Deegan Atha, Frederik Schöller, Sharmita Dey, Seyed Fakoorian, Kyohei Otsu, Barry Ridge, Marko Bjelonic, Lorenz Wellhausen, Marco Hutter, and Ali akbar Agha-mohammadi. Self-supervised traversability prediction by learning to reconstruct safe terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022. 1

[18] Dhruv Shah, Benjamin Eysenbach, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. In *Conference on Robot Learning (CoRL)*, 2021. 4

[19] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A General Navigation Model to Drive Any Robot. In *International Conference on Robotics and Automation (ICRA)*, 2023. 1, 3

[20] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. ViNT: A foundation model for visual navigation. In *7th Annual Conference on Robot Learning*, 2023. 1

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, page 5998–6008, 2017. 7

[22] Lorenz Wellhausen and Marco Hutter. ArtPlanner: Robust legged robot navigation in the field. *Field Robotics*, 3(1):413–434, jan 2023. 1

[23] Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments, 2020. 7

[24] F. Yang, C. Wang, C. Cadena, and M. Hutter. iplanner: Imperative path planning. In *Robotics: Science and Systems Conference (RSS)*, Daegu, Republic of Korea, July 2023. 1